2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications;
Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing

A Dynamic Power Tuning for the Constrained Application Protocol of Internet of Things

Mamoun Qasem, Ahmed Al-Dubai, Imed, Romdhani, *Muneer Bani Yassien
School of Computing, Edinburgh Napier University, UK
*Computer science Department, Jordan University of Science and Technology, Jordan
{M.Qasem, A.Al-Dubai, I.Romdhani}@napier.ac.uk, *masadeh@just.edu.jo

Abstract—**Constrained application protocol (CoAP) has been designed for M2M and IoT applications. The resource directory (RD) in CoAP requires periodical updates by other nodes. However, a regular update process causes extra signaling overhead and drains the node's battery, which consequently reduces the battery lifetime of all nodes. Thus, there is a need to adjust the update interval to prolong the lifetime of the entire network. Although this problem has been investigated in the literature, it is still in its infancy as existing works do not consider the dynamic periodic adjustments of the update messages. To fill in this gap, this paper devise a new adaptive tuning approach between CoAP clients and their CoAP-RD to adjust the frequency of updates based on the battery level of the transmitting node. We evaluated our protocol using simulation experiments. The simulation results confirm that the adaptive tuning approach outperforms its traditional counterpart by 10 to 15% with respect to the overall network lifetime.**

Keywords—Resource Directory (RD), Wireless Sensor, CoAP, IoT, Energy-aware, adaptive.

I. INTRODUCTION

Nowadays, three billion devices are connected to the Internet, and by the end of this decade research studies predict the number to reach up to 50 billion [2]. Internet of Things (IoT) accentuates the second era of Internet evolution in terms of gathering information and providing it as services, where the wireless sensor networks (WSNs) play a vital role to underpin the infrastructure for IoT. However, due to the limitations of sensors such as limited memory capacity, processing, and limited energy, existing standard protocols such as Hypertext Transfer Protocol (HTTP) are not perfectly equipped to cater for supporting efficient Internet Protocol (IP) based applications. Therefore, different IETF working groups have focused on creating new adaptive protocols for the IoT context as shown in Table 1. One of these protocols is the constrained application protocol (CoAP), which has been standardized by the Constrained RESTful Environments (CoRE) working group. CoAP is a web transfer protocol designed particularly for constrained, low-power and lossy networks [5] [20]. This protocol operates in two different modes in terms of resource discovery, centralized and distributed. In the distributed approach the sensor discovers resources offered by another sensor by sending a unicast request message. However, the centralized approach uses resource directory (RD) to host and maintain the updated service information (descriptions) of all nodes within a respective pre-configured domain, and acts on behalf of any node to reply the service request queries.

Table I. PROTOCOLS IN SUPPORT OF THE IOT

Application layer	CoAP	DDS	MQTT	AMOP	MQTT-NS	XMPP	mDNS	DNS-SD
Network layer	RPL							
	6LoWPAN					IPv6/IPv4		
Link layer	IEEE 802.15.4							
Physical layer	IEEE 802.15.4							

CoAP is a client/server protocol similar to and provides also resource discovery mechanism. The client requests certain resources that are already identified by Uniform Resource Identifier (URI), then the RD response by success/failure code. CoAP interchanges messages in an asynchronous manner over UDP and therefore consumes less energy compared to HTTP. In order to keep the RD fully updated, the nodes periodically send update messages along with the latest status of their associated services (for example temperature, humidity, carbon dioxide, etc.). Nevertheless, updating RD drains the nodes batteries and leads increases their failure probability. On one hand, decreasing the update interval helps to maintain an up-to-date service information in the RD, but increases the power consumption of the nodes. On the other hand, increasing this interval may lead to outdated service information and prolongs the node's battery lifetime. An optimal trade-off between the freshness of the updates and the power consumption should be found.

Fig.1 demonstrates clearly the mentioned scenarios [4], where the node B101 battery is depleted without notifying the RD. In this case, node A101 will keep sending request message to B101. In addition, the dead node (B101) will keep receiving requests from the other requesters according to their last update received previously from the RD. In this situation, the quires will not be responded, which dramatically increase unnecessarily the power consumption of the requesting nodes.

978-1-5090-0154-5/15 $31.00 © 2015 IEEE
DOI 10.1109/CIT/IUCC/DASC/PICOM.2015.167

IEEE computer society

To the best of our knowledge this problem has not been investigated in CoAP and existing works did not consider the dynamic adjustments of the update messages [9], [11-13].

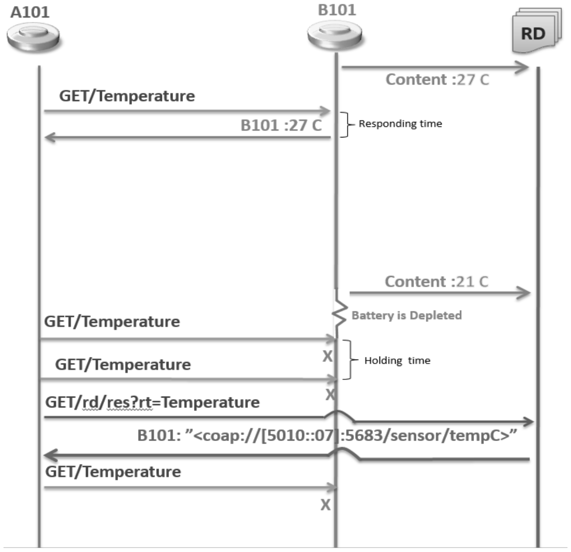

Fig. 1. Failure node demonstration

To address this issue, this paper presents a new adaptive tuning approach to the CoAP protocol. The tuning mechanism takes place between sensor nodes and the resource directory (RD) to adjust the periodicity of updates based on the battery level each sensing node.

The rest of this paper is structured as follows. Section II covers the related work. Section III describes the new adaptive tuning mechanism. The performance of the new approach is illustrated in Section IV. Finally the conclusion and future work are given in Section V.

II. Related Work

Saving power in IoT applications [19] [20] is crucial due to the limited power capacities of sensing nodes. Battery operated nodes, in particular, face the battery replacement issue. Such a replacement is not practical when deployed in hush IoT sensing environment. Many energy efficient protocols have been proposed [14]. Both MicaZ and MSP430 [3] [15] have been designed with less complexity and minimal power consumption. To this end, MSP430 consumes 20 times less power than MicaZ in the sleep mode, and 4 times lower in the active mode.

As reported by IETF, the lightweight protocol CoAP has was designed to reduce the energy consumption using a compact packet overhead [5] [6]. The observe mode keeps the queries overhead to a minimum. Also, in addition to discovery feature where the nodes properties can be discovered within a respective domain [7], [8]. A new Cross layer Optimization Framework attempt to minimize the total power consumption by using cooperative diversity was proposed in [8]. Another technique that uses an exponential backing-off mechanism for the waiting time interval ensures congestion control safety which consequently reduces energy consumption [5]. However, this technique has been used based on the waiting time interval, omitting the battery level, and it is not completely adaptive.

Although many research works have been conducted in the literature under the umbrella of power-saving protocols, the power level of the node's battery is barely considered. Butt et al. in [9] proposed a web services based service discovery protocol to tackle the challenges posed by constrained domains while offering the required interoperability. It is able to provide a service selection technique to offer the appropriate service to the user application depending on the available context information of user and services. However, this scheme hasn't consider the energy levels (i.e the residual power in battery). In this paper, we extend the work of Francesco et al. [10] who proposed an adaptive and cross-layer framework to increase reliability in WSNs based on the IEEE 802.15.4/ZigBee standards. The energy-aware adaptation module grips the application's reliability requirements (in terms of the contention (or congestion) and channel errors) to mitigate the power consumption by tuning the maximum number of retransmissions in MAC layer. However, this approach is limited to the MAC and routing layers and, in particular, the CoAP protocol has not been considered. Therefore, the power consumptions caused by the periodic update messages in CoAP will abridge the node lifetime.

III. The proposed Approach

Our tuning approach aims to tune the frequency of update messages in CoAP based on the current node's battery level. Instead of using a default update frequency interval, the proposed tuning mechanism adjusts the default interval by doubling it to prolong the node's battery lifetime and reduce the network overhead. In the tuning phase, we classify the power level of the node's battery into four levels: high, medium, low, and critical. We assume that each node is able to discover the RD either by state configuration or by a specific announcement and discovery mechanism. We extend the description message between the sender node and the RD to include node's current battery and therefore choose the appropriate update interval.

Table II. The Battery's levels classification

Level of Power	battery level %	Bits equivalent
High	*b>=75	11
Medium	b>=50	10
Low	b>=25	01
Critical	b>=5	00

*b: battery level

When the node's battery level changes, a new interval is considered. Having known the node's battery, the RD can assign services to the nodes in a rational manner. Each battery level is defined by a threshold.

This threshold is expressed by a percentage of the residual energy of the node. For simplicity, the thresholds have been assumed respectively 75, 50, 25 and 5. When advertising its services, a sensor node includes the current level in its service descriptor. Therefore, a new field of 2 bits can be used for this purpose as shown in Table 2. In addition, the node synchronizes and tunes the update frequencies with the RD by including the frequency interval that matches its battery level.

In high power level when (remaining battery level>=75) the RD is updated using the default interval (a standard interval). However, in the medium level when (b>=50) the protocol slows down the update frequency rate and double the default interval (i.e. double the standard interval).When (b>=25), our protocol expands the standard interval to 4 times as shown in Fig. 3 and considers this as low level.

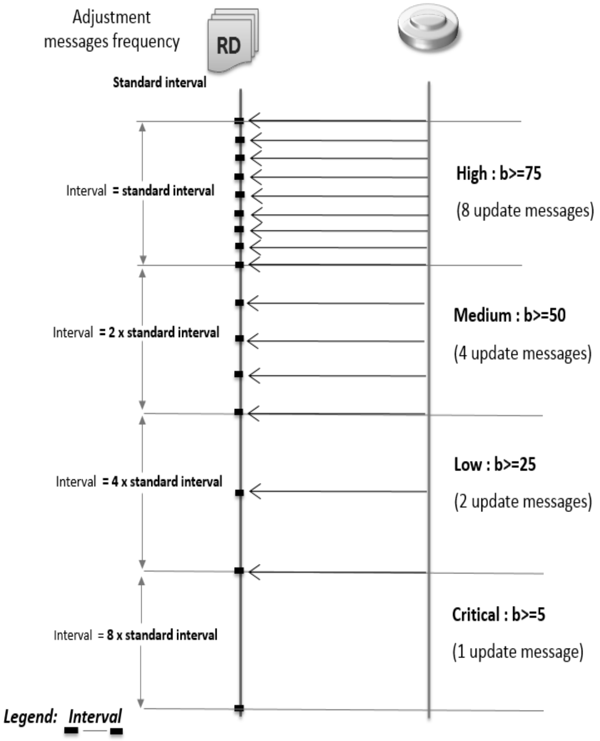

Fig. 2. Adjustment update messages frequency

The last level is the critical level or failover level. When (b>=5) the interval becomes equivalent to 8 standard intervals see Fig. 2. After that, when (b<5) then battery is considered empty and the rest of power (4%) would be enough to deal with a number of emergency quires flagged from specific senders if necessary. The main rules of the proposed approach are given in Algorithm 1.

Algorithm 1: The Dynamic Power Tuning

```
calculate battery_level   // the residual power (1)
if battery_level >=75 then
     WAIT standard_interval then SEND an update of R_info
  else if battery_level >=50 then
     WAIT 2 x (standard_interval) then SEND an update of   R_info
  else if battery_level >=25 then
     WAIT 3 x (standard_interval) then SEND an update of R_info
  else if battery_level >=5 then
     WAIT 4 x (standard_interval) then SEND an update of R_info
  else  SEND: "Node is dead"
end if
```

IV. Performance Evaluation

The performance of tuning and non-tuning techniques has been evaluated and analysed by using the Cooja simulator based on Contiki operating system [16]. Contiki is a lightweight, highly portable open source operating system, it is dedicated for WSNs and widely used for IoT. The simulation parameters are summarized in Table 3.

TABLE III. Simulation configurations for experiments

Parameter	Value
Operating System	Contiki 2.7
Node Type	TMote sky
Routing protocol	RPL
MAC/adaptation layer	ContikiMAC/ 6LowPAN
Radio Environment	Unit Disk Graph Medium (UDGM)
Number of nodes	6-35
Simulation duration	Variable
Full battery	7000 mJ
Transmission range	50, 100, 150,200 (m)
Data packet rate	60 sec

Due to the difficulty of reading the battery level from Cooja to determine the level of power for node's battery, we calculated the power consumption for each node using a function in Contiki called *energest*, then we obtained the residual energy to determine the battery level from equation (1).

$$Residual\ power = Battery - Total\ power\ consumption \qquad (1)$$

In addition, we used TMote sky nodes, so the following operating conditions should be considered to calculate the power consumption: CPU energy consumption, radio receiving power consumption, radio transmitting power consumption, and the power consumption during the sleep mode. Then, we aggregate them together to get the power consumption [17]. Our initial investigations show that the residual energy for the proposed protocol increases with the decrease of battery level. Fig. 5 shows the variation of the node lifetime when the tuning technique is used or not. It can be seen that tuning does not show a clear improvement in saving power to extend the node lifetime at the beginning. However, after reaching the low level (*i.e. when the power level of battery becomes less than 50%*) the notable gain can be recognized. When the node starts with a fully charged battery (ex. 27000 J) our new mechanism can prolong the battery lifetime by approximatively 30 days.

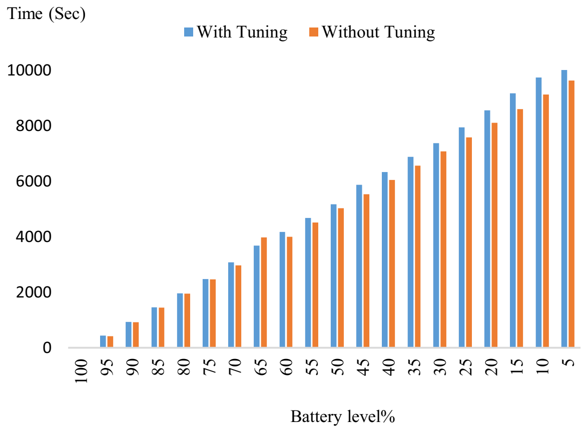

Fig. 5. Battery level consumption behaviour for 10 sec interval

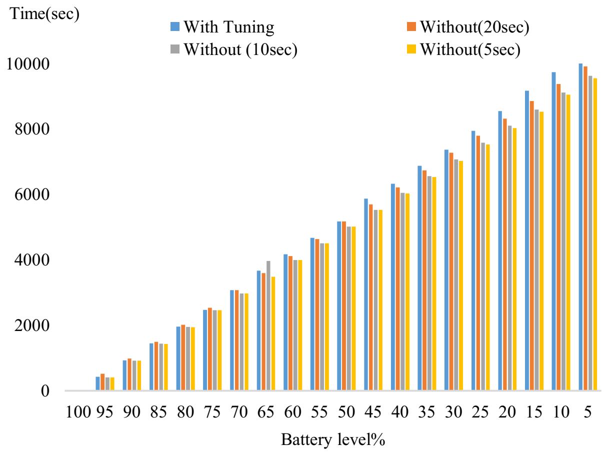

Fig. 6. Battery level consumption behaviour for
4 different intervals

Figure 6 depicts further investigation in terms of varying the timer interval for the non-tuning technique. It is clear that the tuning approach still shows a steady progression even with doubling the non-tuning interval to 20 seconds, or decreasing it to the half (i.e. 5 sec). Figure 7 compares the network lifetime with and without using tuning. We consider the network lifetime based on the first node to die. Results show that when increasing the network size, the network lifetime is decreased because the overhead will increase. However, the tuning approach behaves better than non-tuning approach, and the best results can be observed when the number of nodes is between 5 and 20 as the control overhead in the network is low.

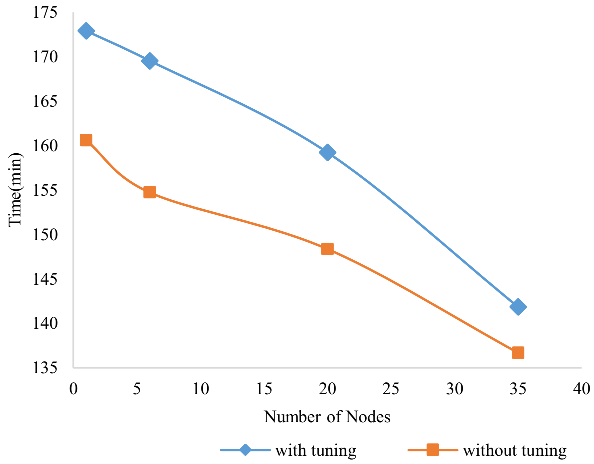

Fig. 7. Energy consumption for different number of nodes

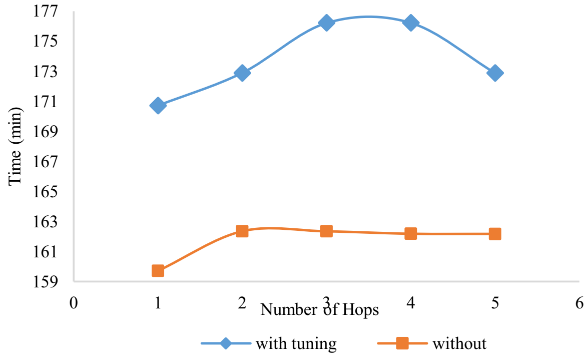

Fig. 8. Energy consumption for different number of hops

Figure 8 plots a steady behavior for the non-tuning approach in terms of the number of hops for 6 nodes. However, the proposed protocol represent better performance with an increase of 10% lifetime, and the best lifetime can be obtained when the number of hops is 3 or 4. In this case the gain rises to 15% and that can be considered in the best tradeoff of our tuning approach. The 6 nodes in Fig. 9 and Fig. 10 behave stably in all situations in terms of power consumption, which means that changing the transmission range and the receiving success ratio (RX) do not affect the power consumption. Accordingly, the difference between both protocols in those figures is 10 sec, and that confirms that the tuning approach has better results in all scenarios.

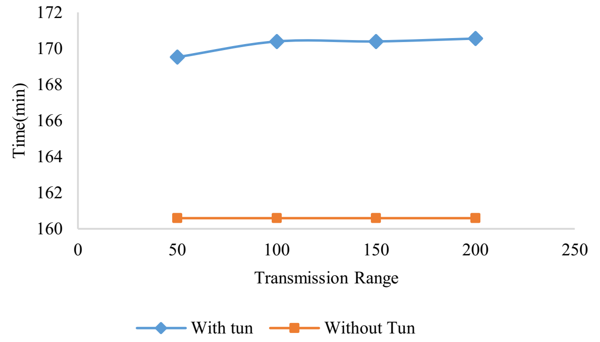

Fig. 9. Energy consumption for different Transmission Range

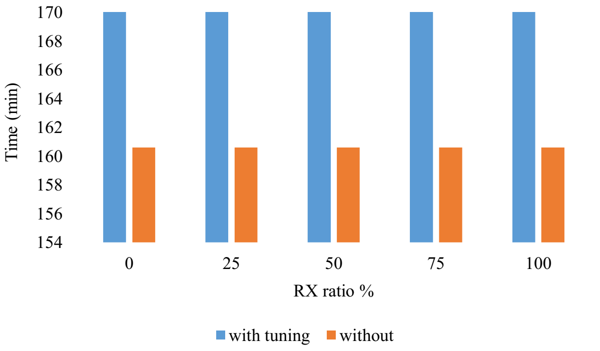

Fig. 10. Energy consumption for different RX ratio

V. Conclusion and Future Work

In this paper, we have proposed an adaptive tuning approach for CoAP to extend the network lifetime. The proposed technique adjusts the frequency of updates messages based on the current residual energy level of each node. The simulation results have shown that the adaptive tuning approach can help to extend the node lifetime by 10%, and the outstanding results can be obtained when the number of hops is 3-4 (i.e. between the sensing node and the RD). In future works, we plan to investigate the failover process when the tuning is deployed. Moreover, we will further analyze the performance of the new technique under a wide range of operating conditions using real testbed and different IoT service discovery protocols.

References

[1] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF, RFC 6550, 2012.

[2] D. Evans, "Cisco IBSG © 2012 Cisco and/or its affiliates" Cisco, 2014. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/IoE.pdf. [Accessed: 30 July 2014].

[3] C.Chauvenet, B. Tourancheau, and D. Catalot, "Energy Evaluations for Wireless IPv6 Sensor Nodes." In SENSORCOMM, The Seventh International Conference on Sensor Technologies and Applications, pp. 97-103. 2013.

[4] B.C Villaverde, R. Alberola, AJ. Jara, S. Fedor, S.K. Das, D. Pesch, "Service Discovery Protocols for Constrained Machine-to-Machine Communications," Communications Surveys & Tutorials, IEEE , vol. 16, no. 1, pp. 41-60, 2014.

[5] Z. Shelby, K. Hartke,C. Bormann, "Constrained Application Protocol (CoAP)," IETF, RFC 7252, June 2014.

[6] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, "Evaluation of constrained application protocol for wireless sensor networks," LANMAN, 18th IEEE Workshop on , pp. 1-6, 2011.

[7] K. Hartke, "Observing Resources in CoAP draft-ietf-core-observe-14," (IETF I-D work in progress), [Expired: in 1st January 2015].

[8] Z. Cao, M. Kovatsch, H. Tian, X. He, "Energy Efficient Implementation of IETF Constrained Protocol Suite draft-ietf-lwig-energy-efficient-00," (IETF I-D work in progress), [Expired: September 22, 2014].

[9] T. A. Butt, I. Phillips, L. Guan, and G. Oikonomou, "Adaptive and Context-Aware Service Discovery for the Internet of Things," in Proc. NEW2AN, pp. 36-47, 2013.

[10] M. Di Francesco, G. Anastasi, M. Conti, S. K. Das, and V. Neri, "Reliability and Energy-Efficiency in IEEE 802.15.4/ZigBee Sensor Networks: An Adaptive and Cross-Layer tpproach," IEEE Journal on Selected Areas in Communications, vol. 29, no. 8, pp. 1508-1524, 2011.

[11] Y. W. e. al, "A framework of energy efficient mobile sensing for automatic user state recognition," In Proc. 7th International Conference on Mobile Systems, Applications, and Services (MobiSys), pp. 179-192, 2009.

[12] Y. Wang; B. Krishnamachari, and M. Annavaram, "Semi-Markov state estimation and policy optimization for energy efficient mobile sensing," SECON, 9th Annual IEEE Communications Society Conference on , pp. 533-541, 2012.

[13] O.Yurur, M. Labrador, and W. Moreno, "Adaptive and Energy Efficient Context Representation Framework in Mobile Sensing," IEEE Trans. on Mobile Computing, vol. 13, no. 8, pp. 1681-1693, 2014.

[14] J. Hill and D. Culle. "Mica: A wireless platform for deeply embedded network," Micro, IEEE , vol. 22, no. 6, pp. 12-24, 2002.

[15] P. Joseph, R. Szewczyk, and D. Culler. "Telos: enabling ultra-low power wireless research," IPSN. 4th International Symposium on , pp. 364-369, 2005.

[16] "Contik O.S and cooja simulatoer" http://www.contiki-os.org/ [Last accessed May2015].

[17] N. Cuartero Moya, S. Quintana Alcaraz, and others, "The MAGCLOUD wireless sensor network," 2011.

[18] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," 2011.

[19] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols and Applications," IEEE Communications Surveys & Tutorials, pp. 1–1, 2015.

[20] C. Anton-Haro and M. Dohler, Machine-To-Machine (M2M) Communications: Architecture, Performance and Applications. Woodhead Publishing Limited, 2015.